BLACK CODHER

CODING PROGRAMME

CODING BLACK FEMALES

Black Codher Bootcamp

NÍ YÓ

BLACK CODHER

CODING PROGRAMME

UNIT 3
JavaScript 101

CODING BLACK FEMALES

NÍ YÓ

WHAT YOU'LL BE LEARNING DURING THIS UNIT

BLACK CODHER
CODING PROGRAMME

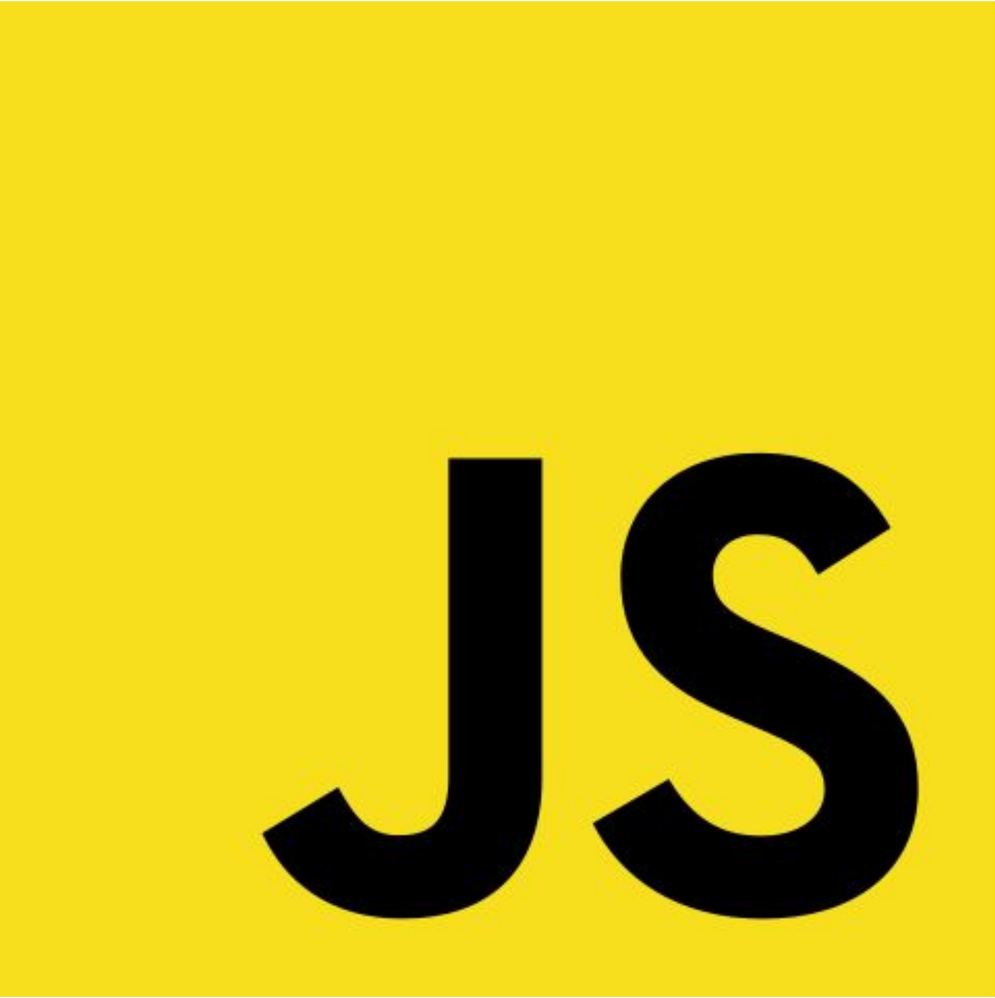
JS

- How to use the Developer tools in Google Chrome (similar approaches work in other browsers)
- How to add interactions to the websites you've already begun building
- How to write JavaScript functionality to create these interactions
- How to interact with the DOM
- How to use a JavaScript library (jQuery)

Copyright © 2020 Black Codher Bootcamp. All Rights Reserved. Do Not Redistribute.

LEARNING OUTCOMES OF THE UNIT

BLACK CODHER
CODING PROGRAMME



- Add interactivity to your Book Shop website
- Building a ToDo list
- Building a Quiz
- More GitHub practice

Copyright © 2020 Black Codher Bootcamp. All Rights Reserved. Do Not Redistribute.

IMPORTANT MESSAGE

BLACK CODHER
CODING PROGRAMME

Programming has some words with special meanings, throughout the material these words will be marked in **bold**. Try your best to remember them, as it will be important that you understand what other programmers mean when they use these words. The material of this course can be used as reference to some of these words.

Copyright © 2020 Black Codher Bootcamp. All Rights Reserved. Do Not Redistribute.

WHAT IS JAVASCRIPT?

```
        }).done(function(response) {
            for (var i = 0; i < response.length; i++) {
                var layer = L.marker(
                    [response[i].latitude, response[i].longitude]
                    // ,{icon: myIcon}
                );
                layer.addTo(group);

                layer.bindPopup(
                    "<p>" + "Species: " + response[i].species + "<
                    "<p>" + "Description: " + response[i].descript
                    "<p>" + "Seen at: " + response[i].latitude + "
                    "<p>" + "On: " + response[i].sighted_at + "</p
            }

            $('select').change(function() {
                species = this.value;
            });
        });
    }
$.ajax({
            url: queryURL,
            method: "GET"
        }).done(function(response) {
            for (var i = 0; i < response.length; i++) {
                var layer = L.marker(
                    [response[i].latitude, response[i].longitude]
                    // ,{icon: myIcon}
                );
                layer.addTo(group);
```

JavaScript is a programming language that runs in all modern web browsers - even when they are offline! It is used to change what is displayed on a web page in response to user activity.

Copyright © 2020 Black Codher Bootcamp. All Rights Reserved. Do Not Redistribute.

WHAT IS IT USED FOR?

Javascript is mainly used to build web applications, but also has other uses. These include:

- Presentations
- Games
- Art
- Server-side Applications
- Web Servers
- Mobile Applications
- Smartwatch Applications
- Robots

...and more

Copyright © 2020 Black Codher Bootcamp. All Rights Reserved. Do Not Redistribute.

BRIEF INTRODUCTION TO ES6+

BLACK CODHER
CODING PROGRAMME

- In 2015 modern JavaScript (ES6) was released
- Key features included:
  - Arrows
  - Let + Const
  - Enhanced object literals
  - Modules
- Since this release there have been yearly updates to JavaScript to include new features
- ES6+ features will be included throughout to prepare you for future employment

Copyright © 2020 Black Codher Bootcamp. All Rights Reserved. Do Not Redistribute.

WELCOME TO THE DOM

BLACK CODHER
CODING PROGRAMME

- The DOM (**D**ocument **O**bject **M**odel) is a visual representation of a web page
- It can be modified with different languages, including Javascript
- The DOM is actually an API (you'll learn more about APIs throughout the course)
- It's one of the most commonly used APIs on the web
- After learning some programming fundamentals you'll be modifying the DOM with your new skills

Copyright © 2020 Black Codher Bootcamp. All Rights Reserved. Do Not Redistribute.

WHAT IS A FRAMEWORK?

A Framework is essentially a library of code for a certain language. Generally, the framework abstracts common tasks and makes it easier and faster for developers to write their specific code. Frameworks don't really do anything by themselves, they just provide an easier platform for people to build on.

Common frameworks include React and JQuery

Copyright © 2020 Black Codher Bootcamp. All Rights Reserved. Do Not Redistribute.

Checkpoint!

How are you feeling?

RED - I have no idea what you're talking about

YELLOW - I have some questions but feel like I understand some things

GREEN - I feel comfortable with everything you've said

Copyright © 2020 Black Codher Bootcamp. All Rights Reserved. Do Not Redistribute.

BLACK CODHER

CODING PROGRAMME

LET THE GAMES BEGIN...

Copyright © 2020 Black Codher Bootcamp. All Rights Reserved. Do Not Redistribute.

WHAT YOU'LL BE LEARNING TODAY?

BLACK CODHER
CODING PROGRAMME

- Chrome developer tools
- `console.log`
- Variables
- Functions

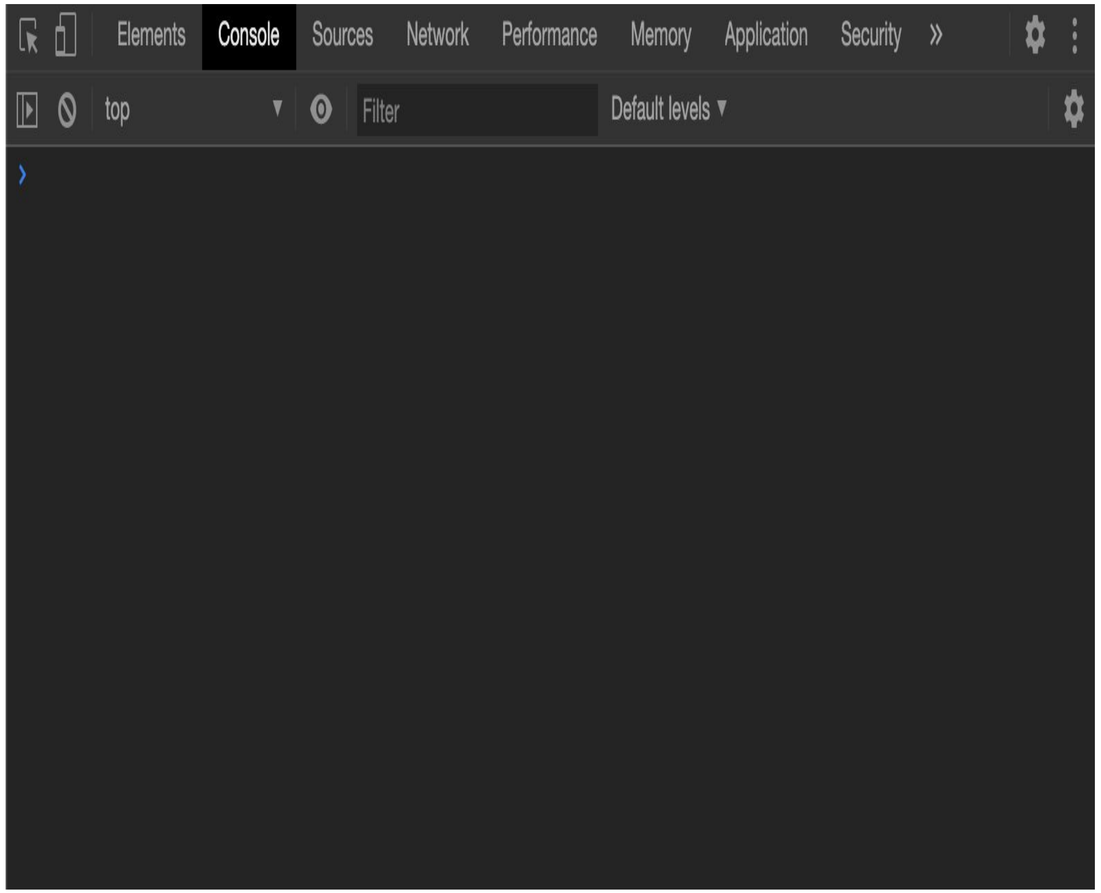
Copyright © 2020 Black Codher Bootcamp. All Rights Reserved. Do Not Redistribute.

WHAT YOU'LL NEED TODAY

BLACK CODHER
CODING PROGRAMME

Google Chrome Dev Tools

- For the first part of this lesson we recommend using the Google Chrome browser, as the instructions have been written to closely match it, however you should be able to so all the same things in any web browser.
- Today, you will also need to open the console.
  - `Ctrl + Shift + J` on *Windows/Linux* or
  - `Alt + Cmd + I` on *Mac*.
- The console should look similar to this:



Copyright © 2020 Black Codher Bootcamp. All Rights Reserved. Do Not Redistribute.

BLACK CODHER

CODING PROGRAMME

CHROME DEV TOOLS

Copyright © 2020 Black Codher Bootcamp. All Rights Reserved. Do Not Redistribute.

CHROME DEV TOOLS

BLACK CODHER
CODING PROGRAMME

- The console can appear in several different places. The next slide has examples of two of them: **"docked to bottom"**, and **"docked to right"**.

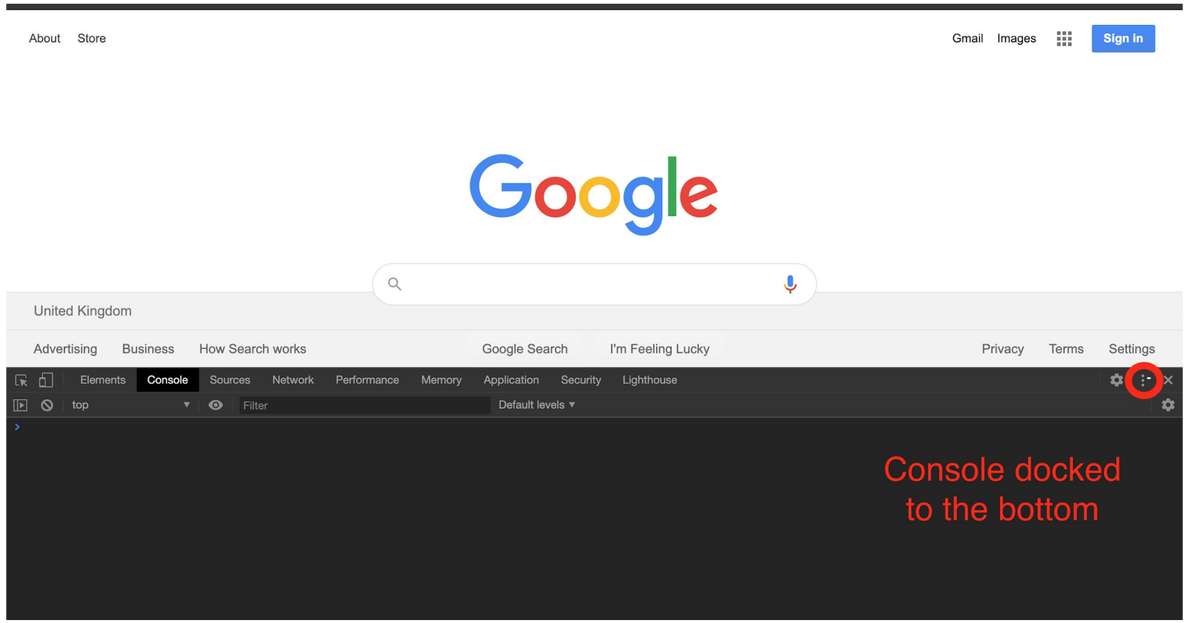
- You can change the location of the console by clicking on the three vertical dots located next to the cross (clicking on the cross will close the console).

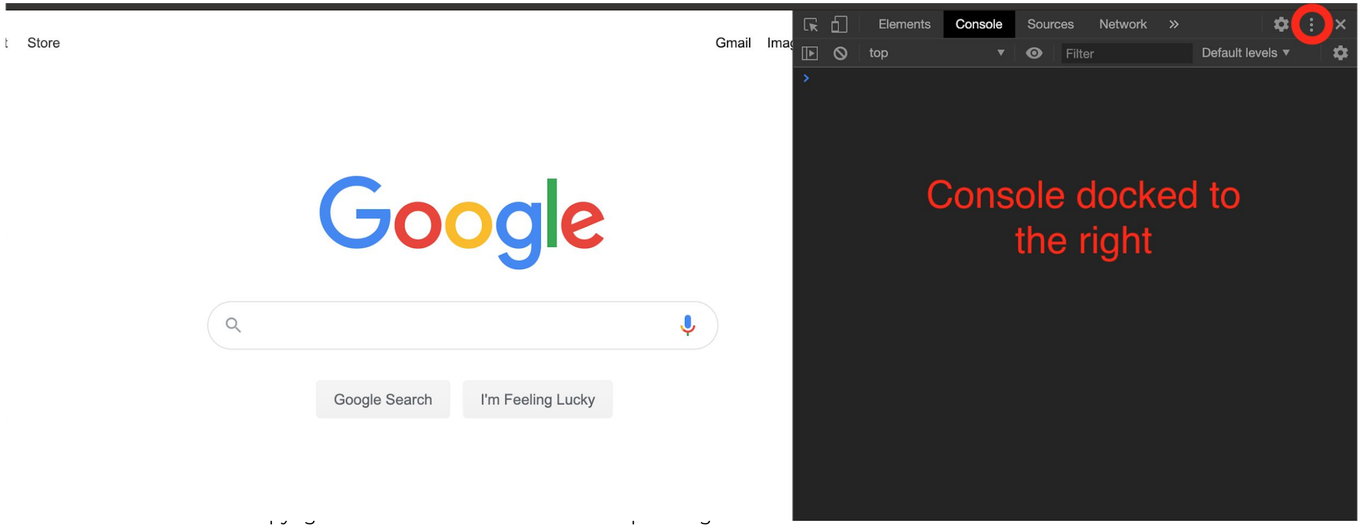
- Their locations are indicated on the pictures in the next slide.

Copyright © 2020 Black Codher Bootcamp. All Rights Reserved. Do Not Redistribute.

CHROME DEV TOOLS

BLACK CODHER
CODING PROGRAMME





LET'S WRITE SOME JAVASCRIPT

BLACK CODHER
CODING PROGRAMME

The console is a place where you can type a single line of JavaScript, and it will immediately run. Try it out now, by typing the following line into the console:

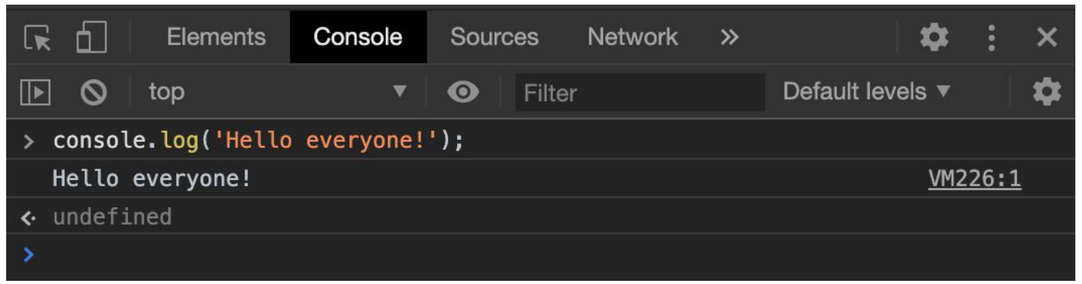
```
console.log('Hello everyone!');
```

Copyright © 2020 Black Codher Bootcamp. All Rights Reserved. Do Not Redistribute.

LET'S WRITE SOME JAVASCRIPT

BLACK CODHER
CODING PROGRAMME

You should now see something like this:



Your screen may not look exactly like the above, but you should see the three lines of text. If you don't see these lines don't worry, just ask for help.

Copyright © 2020 Black Codher Bootcamp. All Rights Reserved. Do Not Redistribute.

WHAT IS A `console.log()`?

BLACK CODHER
CODING PROGRAMME

`console.log` is a JavaScript **function** that takes what you give it, in our case `Hello everyone!`, and prints it to the console.

The other two lines in the console are:

i. The text that we wanted to print out, `Hello everyone!`
ii. `undefined`

More information about `undefined` will be come later in the course.

Copyright © 2020 Black Codher Bootcamp. All Rights Reserved. Do Not Redistribute.

TASK

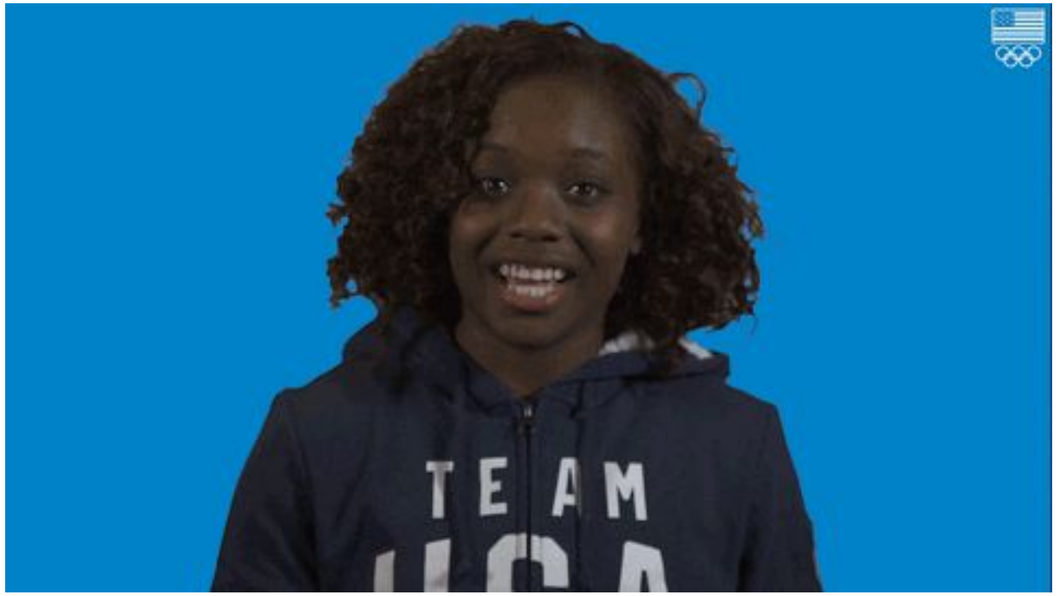
BLACK CODHER
CODING PROGRAMME

- In the folder labelled ***/session1*** you'll find a file called `index.js`
- In this file type the same `console.log` as before
- Save the file
- Reload the page in your browser so you can see the change you made in the console.

You should only see `Hello everyone!` this time, and not other two lines we saw before because you didn't type the JavaScript directly into the console.

Copyright © 2020 Black Codher Bootcamp. All Rights Reserved. Do Not Redistribute.

WHAT YOU'VE LEARNED SO FAR



- How to type JavaScript directly into the browser console
- How to print things to the console with `console.log`
- How to type Javascript in a file
- How to recognise what we see in the console

Copyright © 2020 Black Codher Bootcamp. All Rights Reserved. Do Not Redistribute.

Checkpoint!

How are you feeling?

RED - I have no idea what you're talking about

YELLOW - I have some questions but feel like I understand some things

GREEN - I feel comfortable with everything you've said

Copyright © 2020 Black Codher Bootcamp. All Rights Reserved. Do Not Redistribute.

BLACK CODHER

CODING PROGRAMME

VALUES AND EXPRESSIONS

Copyright © 2020 Black Codher Bootcamp. All Rights Reserved. Do Not Redistribute.

VALUES AND EXPRESSIONS

BLACK CODHER
CODING PROGRAMME

For this part of the lesson we are going to go back to the browser console to type some things.

a. Type `5` into the console. You should see that the number is repeated back to you and with an arrow next to it

b. Type `2 + 2` into the console. You should see the two numbers have been added together and the result is given.

Copyright © 2020 Black Codher Bootcamp. All Rights Reserved. Do Not Redistribute.

VALUES AND EXPRESSIONS

- The `2 + 2` you typed is an **expression**. What is printed in the console on the following line is the **value** of the **expression** you typed.

- In the first example the **value** of `5` is just `5`. In the second example, the `+` indicates that the two numbers should be added together and the result of adding the two numbers together is the **value** of `4`.

Copyright © 2020 Black Codher Bootcamp. All Rights Reserved. Do Not Redistribute.

VALUES AND STRINGS

BLACK CODHER
CODING PROGRAMME

A **value** does not have to just be a number, it can also be a set of letters. In the programming world sets of letters are known as a **string** and are written in quotes.

Type `'Hello'` into the console (with the quotes) and you'll see it repeated back to you.

Copyright © 2020 Black Codher Bootcamp. All Rights Reserved. Do Not Redistribute.

TASK

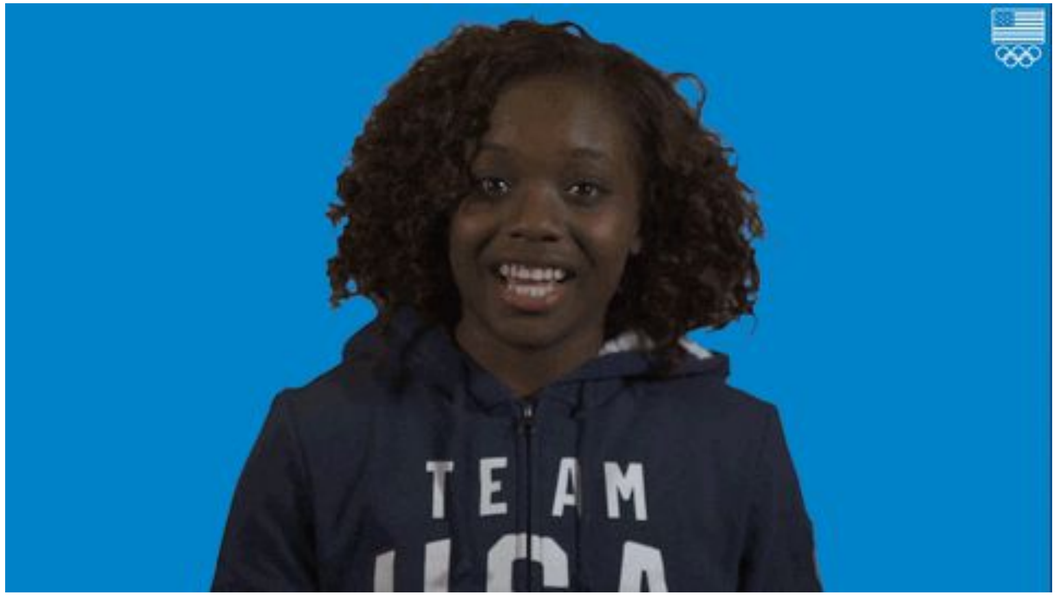
Type the following into the console:

```
'Hey,' + 'how are you?'
```

Ask yourself why adding two string values together could be useful and where you may have seen this being done on the internet?

Copyright © 2020 Black Codher Bootcamp. All Rights Reserved. Do Not Redistribute.

WHAT YOU'VE LEARNED SO FAR



- A **value** can be a single number or a **string**
- An **expression** can combine a several numbers or strings together
- An **expression** can be reduced to a single **value**

Copyright © 2020 Black Codher Bootcamp. All Rights Reserved. Do Not Redistribute.

Checkpoint!

How are you feeling?

RED - I have no idea what you're talking about

YELLOW - I have some questions but feel like I understand some things

GREEN - I feel comfortable with everything you've said

Copyright © 2020 Black Codher Bootcamp. All Rights Reserved. Do Not Redistribute.

BLACK CODHER

CODING PROGRAMME

VARIABLES

Copyright © 2020 Black Codher Bootcamp. All Rights Reserved. Do Not Redistribute.

VARIABLES

Let's go back to the console and type:

```
var a = 85;
```

You should now see the value `undefined` being returned to you.

Copyright © 2020 Black Codher Bootcamp. All Rights Reserved. Do Not Redistribute.

WHAT IS `undefined`?

Using the example that we have written in the console `undefined` means what you have just typed doesn't really have a value.

As we get further into the course you will find that a **method, statement** or **function** can also return `undefined`. More details on those three terms later on in the course.

Copyright © 2020 Black Codher Bootcamp. All Rights Reserved. Do Not Redistribute.

VARIABLES

Whilst what we typed returned `undefined`, it did something else as well and created the **variable** `a`

Copyright © 2020 Black Codher Bootcamp. All Rights Reserved. Do Not Redistribute.

VARIABLES

BLACK CODHER
CODING PROGRAMME

- Back to the console once again, this time just type `a` and you should now see `85` in the console.

- **Variables** allow us to give a name to a **value** for later usage.

In the console type:

```
var b = 5;
var c = a + b;
```

Copyright © 2020 Black Codher Bootcamp. All Rights Reserved. Do Not Redistribute.

VARIABLES

BLACK CODHER
CODING PROGRAMME

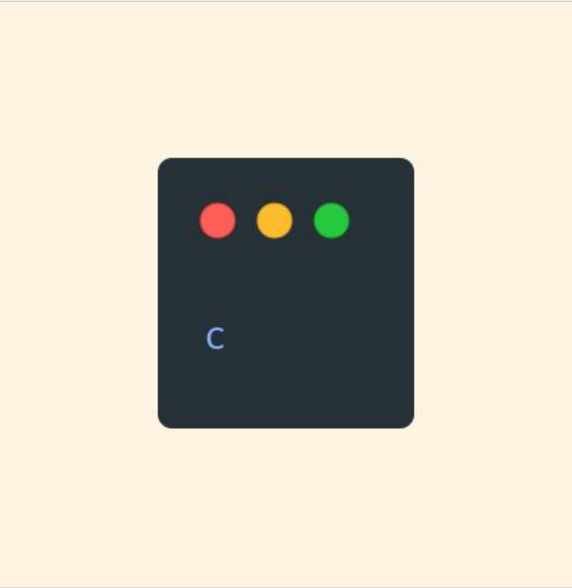
- When starting a new line with var we are creating a new **variable definition**. You should use `var`, `const` or `let` when creating a new variable. We'll cover `const` and `let` later in the course.

- Using `var` we can also change the **value** of a **variable**, hence the name, it can vary.

Copyright © 2020 Black Codher Bootcamp. All Rights Reserved. Do Not Redistribute.

VARIABLES

Returning to the console once again type:



You should see the result: `90`.

Copyright © 2020 Black Codher Bootcamp. All Rights Reserved. Do Not Redistribute.

VARIABLES

Back to the console - we're going to change the value of `c`:

```
b = 10;
```

Notice this time round we don't have to type `var`. If you just have a **variable** name, an equals sign, and an **expression**, then this is a **variable assignment**, which changes the **value** of a **variable** that you defined earlier.

Copyright © 2020 Black Codher Bootcamp. All Rights Reserved. Do Not Redistribute.

VARIABLES

The **value** of `b` has now changed. However, if we type `c` in the console the result is still `90`. That's because we need to reassign `c` to include the new **value** of `b`:

```
c = a + b;
```

The **value** of `c` has now been updated as the **expression** `a + b` is converted into its **value** immediately when used.

Copyright © 2020 Black Codher Bootcamp. All Rights Reserved. Do Not Redistribute.

KEYWORDS AND IDENTIFIERS

BLACK CODHER
CODING PROGRAMME

- `var` is a keyword, meaning it's special to the language.

- Earlier we typed `Hello` into the console without the quotes wrapped around and received an error about it not being defined. You can put anything you like into a **string**, but things that aren't in a **string** need to be defined to mean something before you can use them. The `var` keyword defines a new word to be a **variable**. In this section you created the variables `a`, `b`, and `c` so you could use those.

Copyright © 2020 Black Codher Bootcamp. All Rights Reserved. Do Not Redistribute.

KEYWORDS AND IDENTIFIERS

BLACK CODHER
CODING PROGRAMME

- Unlike with strings, you can't use any word you like to be a variable name. The kind of words that can be variable names are called **identifiers**. **Identifiers** can contain *letters*, *$* signs, or underscores _. They can also have numbers in them, but may not start with a number. The rest of the punctuation on your keyboard cannot be used in identifiers.

- The other important rule is that you can't write an **identifier** that is the same as a keyword, so you cannot create a **variable** called `var`. If you try writing this then you will get a fairly obscure error message.

Copyright © 2020 Black Codher Bootcamp. All Rights Reserved. Do Not Redistribute.

`const`, `let` and `var`

- So far we've been learning about `var`, but there are two other keywords we can use when creating variables `const` and `let`.

- `const`: we use this keyword when we never want to reassign an **identifier**

- `let`: used when we want to indicate a **variable** may be reassigned

- `var`: the variable can either be reassigned or not.

- In modern web development `var` is the least used. For the remainder of the course we will be using `const` and `let`.

Copyright © 2020 Black Codher Bootcamp. All Rights Reserved. Do Not Redistribute.

TASK

BLACK CODHER
CODING PROGRAMME

So far in the examples have used `var`.

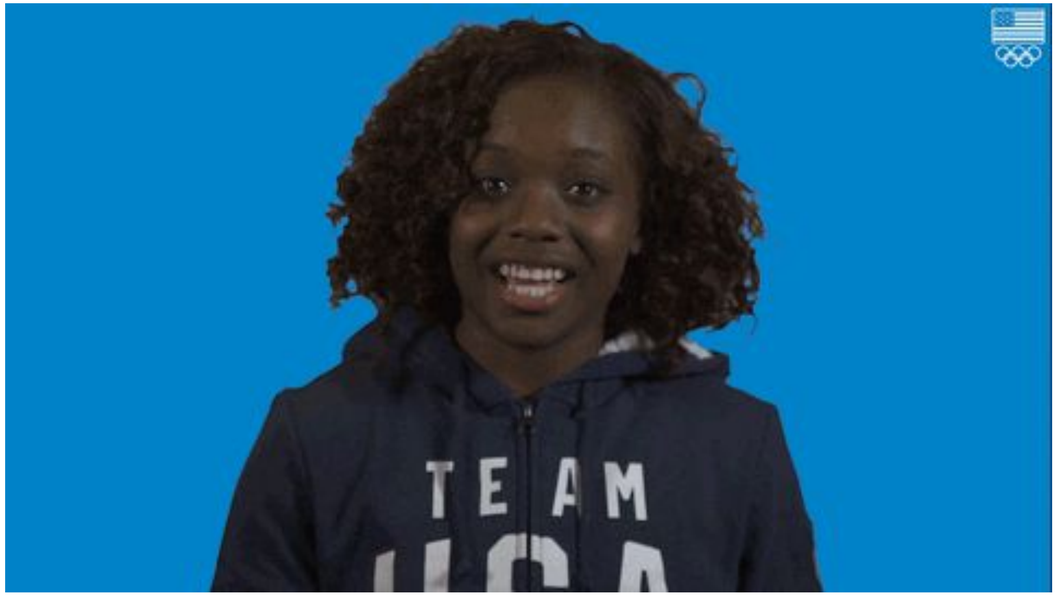
Refresh the browser and convert the variables from the example code to use `const` and `let` instead.

Which variables will need to be `const`, and which will need to be `let`?

If you have time feel free to create more **values** and **expressions** using our `const` and `let` **variables**

Copyright © 2020 Black Codher Bootcamp. All Rights Reserved. Do Not Redistribute.

WHAT YOU'VE LEARNED SO FAR



- What **strings**, **expressions**, **values** and **variables** are
- How to read and write number and string **expressions**
- How to use the `+` operator on number and string expressions
- How to store values in variables
- The differences among `const`, `let` and `var`

Copyright © 2020 Black Codher Bootcamp. All Rights Reserved. Do Not Redistribute.

Checkpoint!

How are you feeling?

RED - I have no idea what you're talking about

YELLOW - I have some questions but feel like I understand some things

GREEN - I feel comfortable with everything you've said

Copyright © 2020 Black Codher Bootcamp. All Rights Reserved. Do Not Redistribute.

BLACK CODHER

CODING PROGRAMME

FUNCTIONS

Copyright © 2020 Black Codher Bootcamp. All Rights Reserved. Do Not Redistribute.

FUNCTIONS

Functions are self contained modules of code that perform a specific task. The can take in data, process it and **return** a result.

In the console type the following:

```
function sayHey() { console.log('Hey!'); }
```

This is a **function declaratio**n. It's recognisable by the keyword **function** at the beginning of it. In the above code we have now created a new **function** called `sayHey`.

Copyright © 2020 Black Codher Bootcamp. All Rights Reserved. Do Not Redistribute.

FUNCTIONS

Let's see what happens when we `call` the function:

```
sayHey();
```

Copyright © 2020 Black Codher Bootcamp. All Rights Reserved. Do Not Redistribute.

FUNCTIONS

BLACK CODHER
CODING PROGRAMME

You should now see `Hey!` in the console. The last piece of code written in the console is a **function call**. We can recognise it because:

- The name `sayHey`
- The parentheses (`()`) immediately after the name

Copyright © 2020 Black Codher Bootcamp. All Rights Reserved. Do Not Redistribute.

FUNCTIONS

Now it's time to move our function into `index.js`. In the console we wrote it all in one line, but normally we write a function this way:

```
function sayHey() {
    console.log('Hey!');
}
```

The first line is the **function signature** and consists of the f**unction name** and the number of **parameters** required (more on **parameters** later in the course). Everything between the `{}` is the function body. Inside the function body we put all the instructions that we want the function to execute. In our example we are saying that we want `sayHey` to print `Hey!` to the console.

Copyright © 2020 Black Codher Bootcamp. All Rights Reserved. Do Not Redistribute.

FUNCTION FLOW OF CONTROL

When a **function** is called (e.g. `sayHey()`) the software program will leave its current section of code and begins to execute the first line inside the body of function. A **function flow** of control goes like this:

1. The program comes to the line of code that contains the **function call**
2. The program enters the **function** starting at the first line of the **function body**
3. All the instructions inside the **function** are carried out from top to bottom
4. The program leaves the **function** and **returns** to the part of the program it was before
5. Any data that was computed and **returned** by the function is used in place of the function in the original line of code

Returned values will be discussed further in the course.

Copyright © 2020 Black Codher Bootcamp. All Rights Reserved. Do Not Redistribute.

TASK

Let's make a bigger **function**. Put the following inside `index.js` underneath our `sayHello` **function**:

```
function conversation() {
    sayHey();
    console.log('How are you?');
    console.log('Goodbye');
}
```

Now call `conversation()` in the console and you will see the order of how the text is printed out.

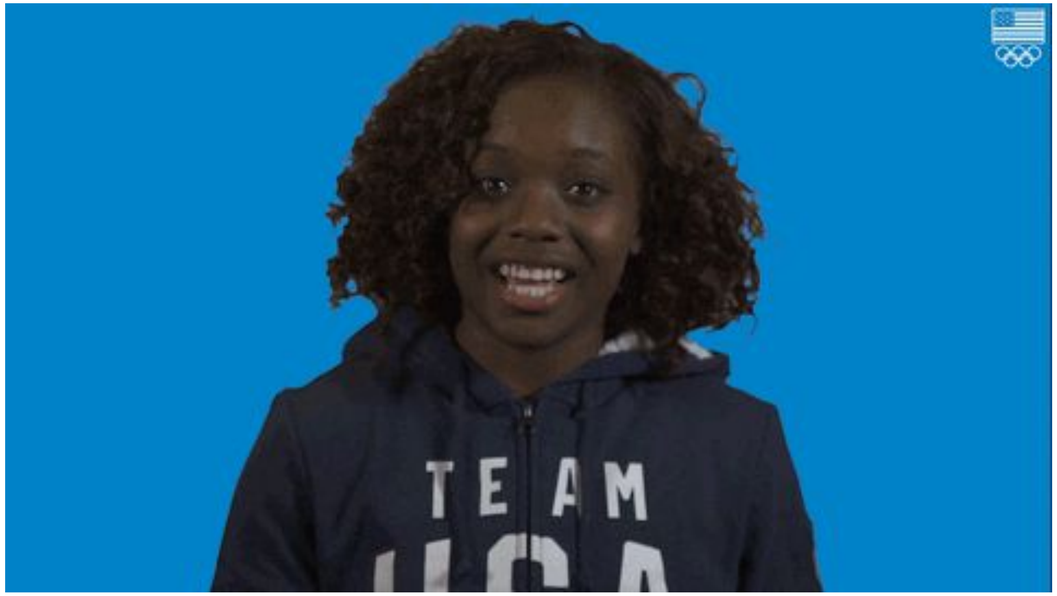
Copyright © 2020 Black Codher Bootcamp. All Rights Reserved. Do Not Redistribute.

FUNCTIONS

One more thing to note, function names are **identifiers**, the same as variables. This means they have the exact same rules about what names you can give to your functions. Remember, the word `function` is a keyword.

Copyright © 2020 Black Codher Bootcamp. All Rights Reserved. Do Not Redistribute.

WHAT YOU'VE LEARNED SO FAR



- What functions are
- How to name functions
- How to define and call functions

Copyright © 2020 Black Codher Bootcamp. All Rights Reserved. Do Not Redistribute.

Checkpoint!

How are you feeling?

RED - I have no idea what you're talking about

YELLOW - I have some questions but feel like I understand some things

GREEN - I feel comfortable with everything you've said

Copyright © 2020 Black Codher Bootcamp. All Rights Reserved. Do Not Redistribute.

BLACK CODHER

CODING PROGRAMME

FUNCTION PARAMETERS

Copyright © 2020 Black Codher Bootcamp. All Rights Reserved. Do Not Redistribute.

FUNCTION PARAMETERS

BLACK CODHER
CODING PROGRAMME

Change the `sayHey` function definition to be:

```
function sayHey(person) {
    console.log('Hey ' + person + '!')
}
```

Now call it with `sayHey('Monique')` (you can also replace `Monique` with your name).

Copyright © 2020 Black Codher Bootcamp. All Rights Reserved. Do Not Redistribute.

FUNCTION PARAMETERS

BLACK CODHER
CODING PROGRAMME

- By placing `person` between the parentheses you have added a **parameter** to the function. Notice when calling the function we are passing a **string** to the function. When the function runs `person` is defined as a new **variable** and the **value** is the name we used.

- Add a `person` parameter to the `conversation` function and see what happens.

Copyright © 2020 Black Codher Bootcamp. All Rights Reserved. Do Not Redistribute.

PARAMETERS VS ARGUMENTS

BLACK CODHER
CODING PROGRAMME

During your career you may hear the word **argument** used instead of **parameter**. In the computer science there are subtle differences between the two, but in practice they are used interchangeably.

Copyright © 2020 Black Codher Bootcamp. All Rights Reserved. Do Not Redistribute.

FUNCTION RETURN VALUES

BLACK CODHER
CODING PROGRAMME

Make a new function called greeting with a person parameter.

In the function body write the following:

```
return 'Hey ' + person + '!';
```

When you call this function from the console you will find that unlike our previous functions, this one has a value.

Copyright © 2020 Black Codher Bootcamp. All Rights Reserved. Do Not Redistribute.

FUNCTION RETURN VALUES

The keyword `return` simply means *return the value of this expression from this function*. The value of a function call is the value that is given to `return`. If a function gets to the end of the body and doesn't see the `return` keyword then the value of the call is `undefined`.

Copyright © 2020 Black Codher Bootcamp. All Rights Reserved. Do Not Redistribute.

TASK

Change the `sayHey` function to call the new `greeting` function instead of the `console.log` we have in there. Does `conversation` still work?

Copyright © 2020 Black Codher Bootcamp. All Rights Reserved. Do Not Redistribute.

MULTIPLE PARAMETERS

Functions can have as many parameters as you want, you just need to separate them with commas. Try adding the following two **parameters** to `conversation` :

- `person`
- `topic`

Also change the function body to print the below to the console:

```
"Do you like " + topic + "?"
```

Copyright © 2020 Black Codher Bootcamp. All Rights Reserved. Do Not Redistribute.

MULTIPLE PARAMETERS

Now call `conversation` (similarly to the below) and replace the `person` and `topic` parameters with ones of your choice:

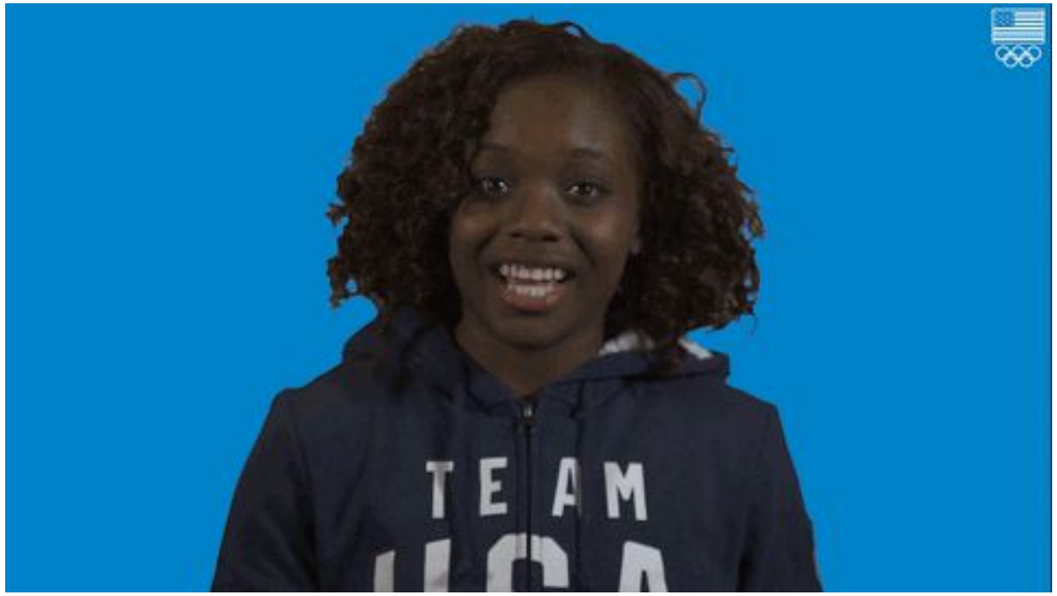
- `person`
- `topic`

Also change the function body to print the below to the console:

```
conversation('Monique', 'pizza');
```

Copyright © 2020 Black Codher Bootcamp. All Rights Reserved. Do Not Redistribute.

WHAT YOU'VE LEARNED SO FAR



- How to add parameters to functions
- How to return a calculated value from a function

Copyright © 2020 Black Codher Bootcamp. All Rights Reserved. Do Not Redistribute.

Checkpoint!

How are you feeling?

RED - I have no idea what you're talking about

YELLOW - I have some questions but feel like I understand some things

GREEN - I feel comfortable with everything you've said

Copyright © 2020 Black Codher Bootcamp. All Rights Reserved. Do Not Redistribute.

BLACK CODHER

CODING PROGRAMME

MORE FUN-FUN-FUNCTIONS

Copyright © 2020 Black Codher Bootcamp. All Rights Reserved. Do Not Redistribute.

FUNCTION EXPRESSIONS

A **function expression** is very similar and has almost the same syntax as a **function declaration**. The main difference between a **function expression** and a **function declaration** is the **function** name, which can be omitted in **function expressions** to create anonymous **functions**

Copyright © 2020 Black Codher Bootcamp. All Rights Reserved. Do Not Redistribute.

FUNCTION EXPRESSIONS

```
const x = function(y) {
    return y * y;
}
```

In this example of a **function expression**, the results of the function will automatically be assigned to the variable x

Copyright © 2020 Black Codher Bootcamp. All Rights Reserved. Do Not Redistribute.

THE FUNCTION CONSTRUCTOR

BLACK CODHER
CODING PROGRAMME

Constructors are like regular functions, but we call them with the **new** keyword.

A **constructor** is useful when you want to create multiple similar objects (we'll be covering objects later in the course) with the same properties and methods.

It's a convention to capitalize the name of constructors to distinguish them from regular functions.

Copyright © 2020 Black Codher Bootcamp. All Rights Reserved. Do Not Redistribute.

THE FUNCTION CONSTRUCTOR

BLACK CODHER
CODING PROGRAMME

Let's take a look at the following code:

```
function Book() {
    // code
}

const myBook = new Book();
```

The last line of the code creates an instance of `Book` and assigns it to a variable. Although the `Book` **constructor** doesn't do anything, `myBook` is still an instance of it. As you can see, there is no difference between this function and regular functions except that it's called with the **new** keyword and the function name is *capitalised*.

Copyright © 2020 Black Codher Bootcamp. All Rights Reserved. Do Not Redistribute.

HOISTING

BLACK CODHER
CODING PROGRAMME

In JavaScript **hoisting** is a mechanism where **variables** and **function declarations** are moved to the top of their scope before code execution.

In simpler terms, no matter where **functions** and **variables** are declared, they are moved to the top of their **scope** regardless of whether their **scope** is **global** or **local**.

Copyright © 2020 Black Codher Bootcamp. All Rights Reserved. Do Not Redistribute.

HOISTING

BLACK CODHER
CODING PROGRAMME

	var	const	let
scope	global or local	block	block
redeclare?	yes	no	no
reassign?	yes	no	yes
hoisted?	yes	no	no

Linkedin

Copyright © 2020 Black Codher Bootcamp. All Rights Reserved. Do Not Redistribute.

IMMEDIATELY INVOKED FUNCTION EXPRESSIONS (IIFE)

BLACK CODHER
CODING PROGRAMME

An **Immediately-invoked Function Expression** is a way to execute functions immediately, as soon as they are created. **IIFEs** are very useful because they don't pollute the **global object**, and they are a simple way to isolate **variables declarations**.

Copyright © 2020 Black Codher Bootcamp. All Rights Reserved. Do Not Redistribute.

IMMEDIATELY INVOKED FUNCTION EXPRESSIONS (IIFE)

BLACK CODHER
CODING PROGRAMME

```
(function() {
 // code
}())
```

Essentially, we have a function defined inside parentheses, and then we append `()` to execute that **function**: `(/* function */)()`.

Those wrapping parentheses are actually what make our **function**, internally, be considered an **expression**.

Copyright © 2020 Black Codher Bootcamp. All Rights Reserved. Do Not Redistribute.

ARROW FUNCTIONS

BLACK CODHER
CODING PROGRAMME

Arrow functions were first introduced in ES6 and allow us to write shorter function syntax.

Before

```
const hey = function() {
    return 'hey ladies!';
}
```

With arrow function

```
const hey = () => {
    return 'hey ladies!';
}
```

Copyright © 2020 Black Codher Bootcamp. All Rights Reserved. Do Not Redistribute.

ARROW FUNCTIONS

With arrow functions we can make this even shorter. *If* the function has only one statement, and the statement returns a value, you can remove the brackets *and* the `return` keyword:

```
const hey = () => 'hey ladies';
```

Copyright © 2020 Black Codher Bootcamp. All Rights Reserved. Do Not Redistribute.

ARROW FUNCTION PARAMETERS

BLACK CODHER
CODING PROGRAMME

If you have parameters they can be passed inside the parenthesis:

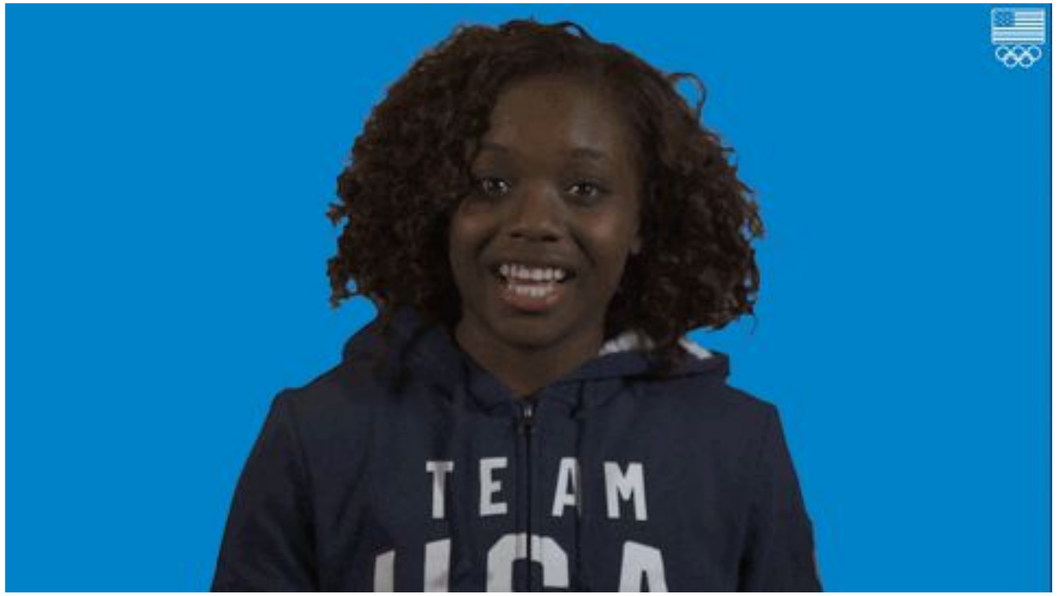
```
const hey = (value) ⇒ 'Hey ' + value;
```

And if you only have one parameter you can completely skip the parenthesis:

```
const hey = value ⇒ 'Hey ' + value;
```

Copyright © 2020 Black Codher Bootcamp. All Rights Reserved. Do Not Redistribute.

WHAT YOU'VE LEARNED SO FAR



- How to add parameters to functions
- How to return a calculated value from a function

Copyright © 2020 Black Codher Bootcamp. All Rights Reserved. Do Not Redistribute.

Checkpoint!

How are you feeling?

RED - I have no idea what you're talking about

YELLOW - I have some questions but feel like I understand some things

GREEN - I feel comfortable with everything you've said

Copyright © 2020 Black Codher Bootcamp. All Rights Reserved. Do Not Redistribute.

BLACK CODHER

CODING PROGRAMME

SUMMARY

Copyright © 2020 Black Codher Bootcamp. All Rights Reserved. Do Not Redistribute.

SUMMARY

BLACK CODHER
CODING PROGRAMME

You now know what all of these words mean:

- strings
- expressions
- values
- variables
- functions (definitions and calls)
- parameters

Copyright © 2020 Black Codher Bootcamp. All Rights Reserved. Do Not Redistribute.

SUMMARY

BLACK CODHER
CODING PROGRAMME

You also know how to do all these things:

- Use the javascript console
- Store values in variables
- Add numbers and combine strings with `+`
- Define and call functions
- Understand what everything in `console.log('Hello!');` means

Copyright © 2020 Black Codher Bootcamp. All Rights Reserved. Do Not Redistribute.